

How to talk to your programmer

Janelle Jenstad
Martin Holmes
</endings>

JJ </e> MH



Finding a programmer

- Institution has tame programmers whose time you can prevail on.
- Project has money to hire a programmer.
- You have a friend or colleague who is a programmer.



Tame institutional programmers

- Will impose institutional standards and requirements on code and output.
 - Standards are good.
 - Long-term support is good.
 - You may not get exactly what you want.



Hired gun programmers

- You have complete control (in theory).
- You are at their mercy (in reality).
- They have no interest in long-term reliability or standards-compliance.
- When it breaks, they aren't there to fix it. Unless you pay them again. Which means they have a vested interest in its breaking.



Friend or colleague

- Professional programmer?
 - If so, doing it as a favour off the side of the desk.
 - If not, may make elementary mistakes that lead to disaster.
 - If you're unhappy, you have a diplomatic issue on your hands.



One last thing...

- DON'T hire your best friend's son who writes websites in his bedroom for \$10 an hour. DON'T DO THAT.



What motivates a programmer?

- Money
- New Cool and Exciting stuff (NCE)
- Challenges
- Pride and ownership



Money

- You probably don't have enough to make a difference.
- If it's the primary motivation, then the programmer has an incentive to finish ASAP. Corners will be cut.



New Cool and Exciting stuff

- THIS IS DANGEROUS:
 - They obviously don't know it well because it's new, cool and exciting.
 - It will be buggy and fragile, because it's N... C... E...
 - It probably won't last. Most things don't, especially NCE things.



Challenges

- This one is your friend. Programmers love to solve difficult problems or devise ingenious solutions to new obstacles.
- Convince the programmer your project has unique components and challenges never before addressed satisfactorily.
- How do you do that?



Creating challenges

- Find examples of existing sites/projects which:
 - Show the general approach you want to take
 - Are open-source, well-documented, and admired
 - Have specific weaknesses you can point at and say “We want to do a better job here.”



Pride and ownership

- This is the most important thing.
- Most programmers work for free on projects they care about.
- Building a career/reputation requires a portfolio of projects for which you can claim ownership and responsibility.
- Bring the programmer into the project family.
- How can you do that?



Adopting a programmer

- Have their code in the same repo as your data.
- Have them attend project meetings.
- Get them doing other project work (encoding, for example; they will quickly invent really helpful shortcuts if they have to do it themselves).
- Credit them from the beginning on the site and in the documentation.
- Involve them in presentations, publications, social media etc.



Dos and don'ts

JJ </e> MH



1. Learn stuff (1)

- Learn as much as you can before looking for a programmer.
- Look at lots of projects like yours.
- Find out how they work.
- Learn all the acronyms (XML, XSLT, PHP....)
- Learn how those technologies fit together.



1. Learn stuff (2)

- Don't despise such knowledge as beneath you.
- Conversely, don't assume it's beyond you. You have a PhD.
- Identify old, stable technologies you can trust.
- Identify technologies supported at your institution.
- Identify respected, stable international standards.



2. If you're hiring

- Identify candidates based on previous work.
- Talk as much as possible before hiring.
- Make your requirements very clear.
- Don't get intimidated by technical talk. If they won't explain what/why/how, don't hire them.
- Warning sign: "You just leave that to me..."



3. Create a stimulating challenge

- Make stability and longevity into a challenge, with these clear goals:
 - Minimal dependencies.
 - Standards-compliance and validation of EVERYTHING
 - Longevity = 50 years.
 - Site should work without a server.
- Point them at Endings and JamStack.org.



Maintain a good working relationship

- Ask their advice and listen to it.
- Check in with them regularly. (Don't have them do 50 hours of coding then say "Oh. That's not what I meant at all.")
- Don't keep changing your mind about what you want.

